УДК 004.056:004.057.3

DOI: 10.18372/2073-4751.75.18014

Гнатюк С.О., д.т.н.,
orcid.org/0000-0003-4992-0564,
Бердибаєв Р.Ш., PhD,
orcid.org/0000-0002-8341-9645,
Богун А.М.,
orcid.org/ 0000-0003-2049-2260,
Сидоренко В.М., к.т.н.,
orcid.org/0000-0002-5910-0837,
Положенцев А.А.,
orcid.org/0000-0003-0139-0752,
Жигаревич О.К.,
orcid.org/0000-0002-7154-9733

ІНТЕГРАЦІЙНА ШИНА ДАНИХ ДЛЯ ЕФЕКТИВНОГО ФУНКЦІОНУВАННЯ СИСТЕМИ УПРАВЛІННЯ ПОДІЯМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Національний авіаційний університет

s.gnatyuk@nau.edu.ua,
abogun@ukr.net,
v.sydorenko@ukr.net,
artem.polozhencev@gmail.com

Вступ

У сучасному світі спостерігається постійний приріст кількості кіберзагроз, що пояснюється зростанням нових інформаційно-комунікаційних технологій (ІКТ) та недостатньою перевіркою розробленого програмного та фізичного забезпечення, а також забуттям про обслуговування та підтримку застарілого програмного та серверного забезпечення. Крім того, постійно з'являються нові уразливості у програмному забезпеченні та архітектурі електронних пристроїв, що знижують рівень кібербезпеки. Для вирішення цих проблем та зниження впливу загроз було створено багато різноманітних інструментів, одним з яких є системи управління подіями інформаційної безпеки, або *Security Information and Event Management* (*SIEM*) [1]. Основною метою сучасної *SIEM* системи є запобігання можливим наслідкам використання уразливостей зловмисниками та мінімізація збитків для користувачів [2].

Аналіз останніх досліджень та постановка завдання

У роботі [3] було запропоновано базову концепцію хмарної архітектури *SIEM* (рис. 1) для різних секторів критичної інфраструктури (КІ). Ця схема також може бути інтегрована в реальні ІКТ-інфраструктури з існуючими *SIEM* (пропонованими різними постачальниками [2]) та іншими інструментальними засобами управління інцидентами. Основними структурними одиницями запропонованої *SIEM* системи є наступні:

- Горизонтальні бази даних (*Horizontal Databases*);
- Блоки аналітики та моніторингу (*Blocks of Analytics and Monitoring*);
- Хмарне зберігання (*Cloud storage*);
- Шифратор (*Encryptor*);
- Брокер повідомлень (*Message Broker*);
- Джерела (Система 1 – Система *N*) *Sources* (*System* 1 – *System N*).

Одним із суттєвих компонентів даної системи є шифратор, який створює єдине блочне хмарне сховище для забезпечення конфіденційності необроблених даних після їх збору за допомогою *syslog*, *NetFlow* тощо. Крім того, *Virtual Box* передає зібрані та зашифровані дані у горизонтальні бази даних через брокерів повідомлень.

У випадку відсутності зв'язку з брокером, тимчасове зберігання даних здійснюється у хмарному сховищі. Дослідження [2] зосереджувалося на аналізі сучасних *SIEM* систем, у [3] розробляли базову концепцію хмарної *SIEM* системи, тоді як [4] описує типи баз даних у контексті реалізації *SIEM* систем.

Крім того проведений аналіз показав, що використання інтеграційної шини даних (*Enterprise Service Bus, ESB*), як складового компонента *SIEM* системи, дасть можливість аналізувати дані у будь-яких форматах та переводити їх у зручний стандартизований вигляд. Отже, як наслідок, метою цієї статті є розробка *ESB* для ефективного функціонування *SIEM* систем на об'єктах КІ.

Особливості архітектури та порівняльний аналіз сучасних ESB

Основою будь-якої *Service Oriented Architecture* (*SOA*) архітектури є *ESB*. *SOA* надає можливість багаторазового використання програмних компонентів за допомогою сервісних інтерфейсів (рис. 2) [5]. Такі інтерфейси використовують загальні комунікаційні стандарти, тому їх можна швидко інтегрувати в нові додатки без необхідності кожного разу проводити глибоку інтеграцію.

SOA-сервіс включає код та інтеграцію даних, необхідні для виконання певної бізнес-функції [6]. Інтерфейси сервісів мають низький рівень взаємозалежності, що дозволяє використовувати їх навіть з мінімальними знаннями про інтеграційний процес.

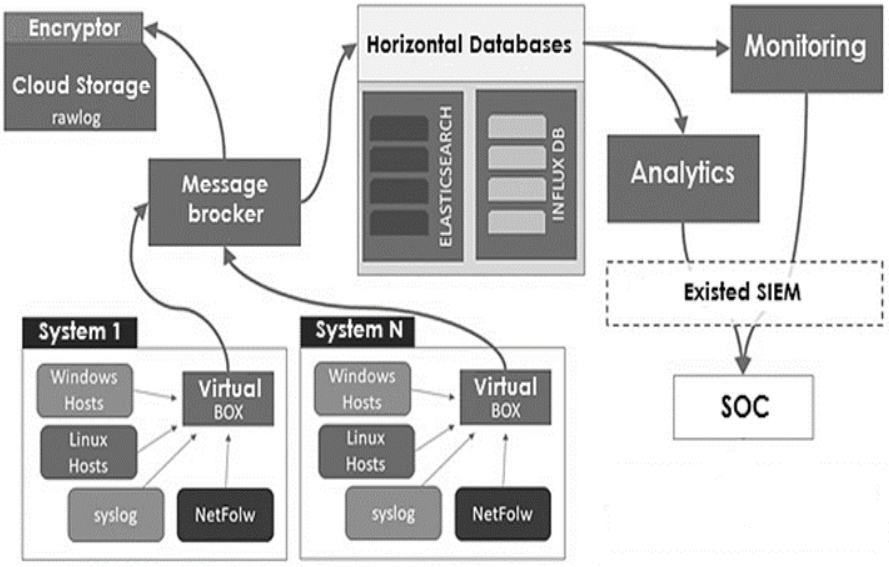

Рис. 1. Запропонована концепція хмарної архітектури *SIEM*

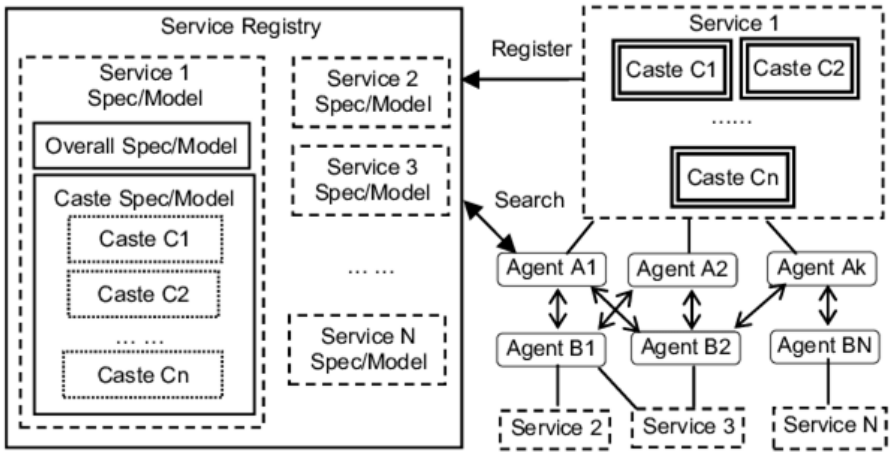

Рис. 2. Схема побудови *SOA*

Доступ до сервісів здійснюється через стандартні мережеві протоколи, такі як *SOAP* / *HTTP* або *JSON* / *HTTP*, що надсилають запити на читання / модифікацію даних. Сервіси публікуються таким чином, щоб розробники могли швидко знаходити їх та повторно використовувати для створення нових додатків. Ці сервіси можуть бути створені як з нуля, так і експортовані з існуючих систем у вигляді інтерфейсів.

В архітектурі *SOA*, сервіси можуть взаємодіяти незалежно від свого типу. Це означає, що конкретний сервіс може бути специфічним для платформи або протоколу, але *SOA* дозволяє таким сервісам взаємодіяти і обмінюватися даними. Обмін цими даними здійснюється через *ESB* [7], що є основою будь-якої архітектури *SOA*. Таким чином, *ESB* виступає шаблоном (рис. 3), в якому централізований компонент інтегрується з основними системами і отримує доступ до цих інтеграцій через сервісні інтерфейси.

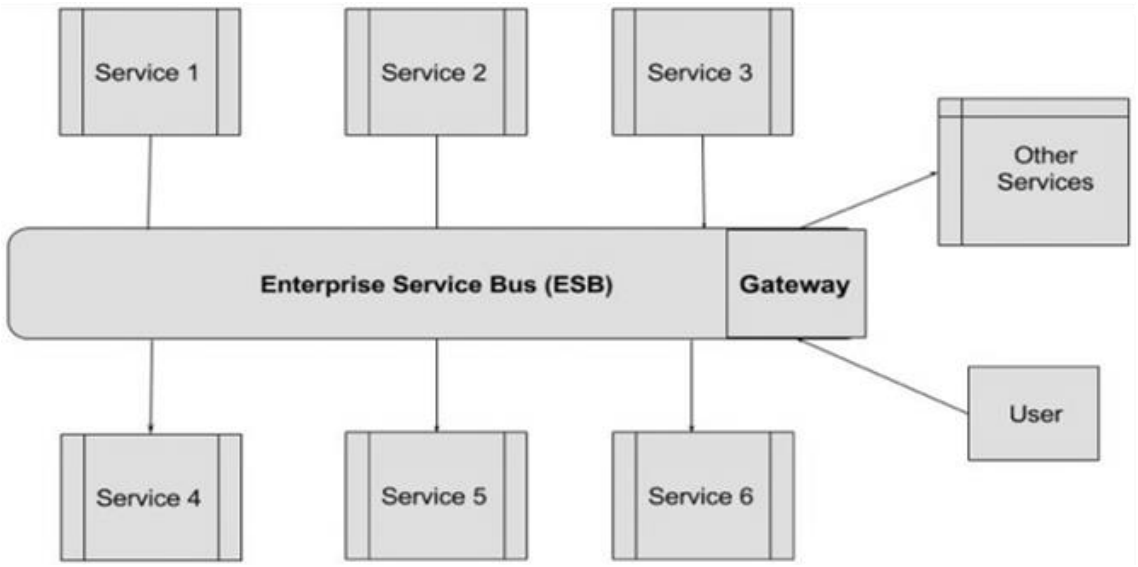

Рис. 3. Схема побудови *ESB*

Крім того, *ESB* дозволяє здійснювати перетворення моделей даних, тісну взаємодію, маршрутизацію і створення декількох запитів, об'єднуючи ці функції в єдиному сервісному інтерфейсі, який може багаторазово використовуватися новими додатками. Шаблон *ESB* зазвичай реалізується за допомогою спеціально розробленого середовища виконання інтеграції та інструментів, що ефективно виконують вищезазначені функції [8].

Загалом, *SOA* може бути реалізована і без *ESB* (рис. 4). Однак, в такому випадку, власникам додатків доведеться знайти унікальний спосіб надання доступу до інтерфейсів, що є дуже трудомістким завданням, особливо за наявності декількох інтерфейсів. Крім того, це може ускладнити подальше обслуговування.

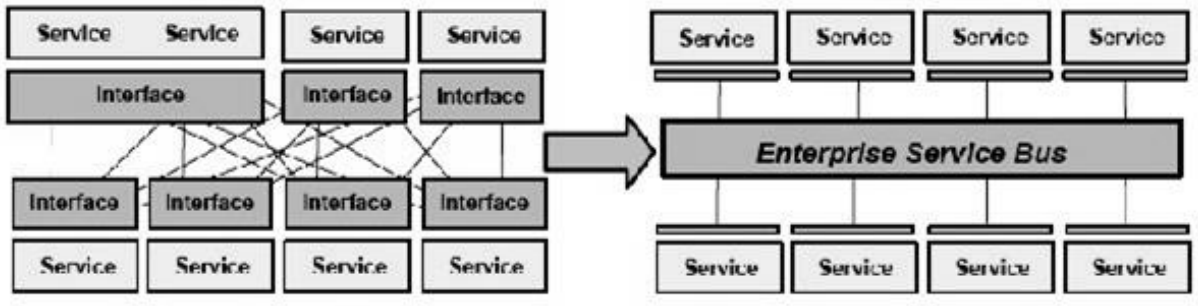

Рис. 4. Порівняння *SOA* з *ESB* та без нього

Також, *ESB* можна реалізувати, використовуючи сервери *JMS* і *XML*/*XSD* як засоби передачі даних між різними сервісами. Різні сервіси реєструються або підключаються до цих *JMS*-серверів і обмінюються даними у форматі *XML*. До набору *SOA* також входять адаптери, що допомагають конвертувати повідомлення у формат, зрозумілий сервісу і *XML*, і навпаки.

Для прикладу розглянемо реалізацію системи біржової торгівлі. Повідомлення з біржі надходять за протоколом *FIX*. Але за архітектурою додаток очікує повідомлення у форматі *JSON*. В такому випадку, можна використати підхід *SOA*, де *FIX*-адаптер перетворить *FIX*-повідомлення в *XML*, а потім це повідомлення передається *JSON*-адаптеру через *ESB*. *ESB* виконує перетворення в *JSON*, необхідний для роботи з кінцевою системою. На рис. 5 відображено приклад реалізації *JBoss ESB* [6], який демонструє такий підхід.

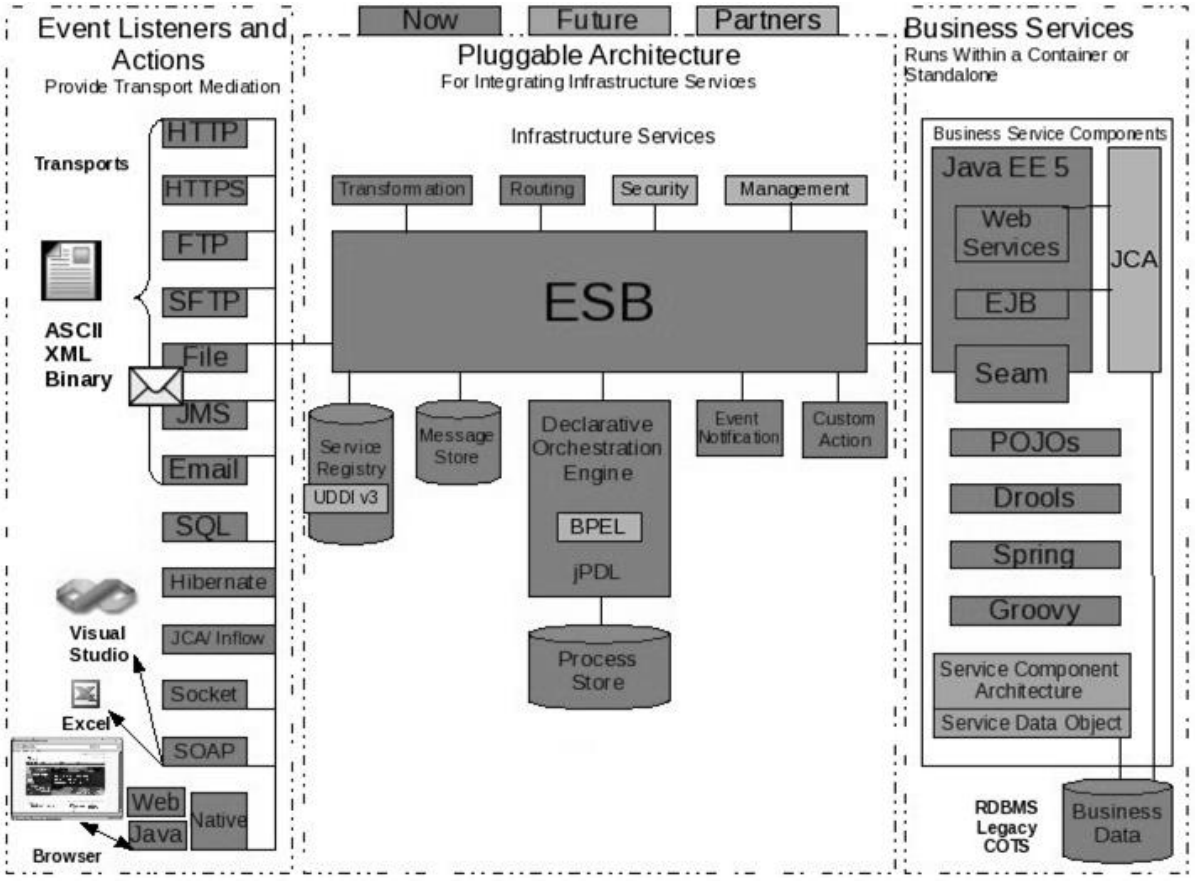

Рис. 5. Приклад реалізації *JBoss ESB*

Особливості сучасних ESB

Інтеграційна шина даних – це програмне забезпечення, яке працює як центральний механізм обміну повідомленнями між інформаційними системами та додатками. Шина має наступні основні принципи:

1. Модернізація елементів вимагає масштабної переробки інтерфейсу. Наприклад, оновлення бази даних *Oracle* потребує переробки всіх пов'язаних інтеграцій.

2. Логування подій реалізовано порізному у кожній інтеграції. Це може ускладнити виявлення і виправлення помилок або втрати даних.

3. Для додавання нових елементів системи потрібні значні інвестиції в інтеграцію *Point-to-Point*. Наприклад, додавання нових торгових майданчиків вимагатиме інтеграції з інтернет-магазином, *CRM, WMS, ERP, PIM* та іншими.

4. Бізнес-аналітика ускладнюється через різні джерела та формати даних. Об'єднання даних для прийняття управлінських рішень стає складним.

5. Збільшення інфраструктури збільшує час і ресурси, необхідні для підтримки системи, знижуючи резерв ресурсів для подальшого покращення.

6. *ESB*-шина об'єднує кілька функцій, які в інтеграціях реалізовано окремо.

7. *ESB*-шина збирає інформацію з різних систем (внутрішніх та зовнішніх) – у форматі, в якому вона міститься в системі-джерелі.

8. Дані шини перетворюються в необхідні формати для передачі в інші системи.і

9. Оператор задає логіку маршрутів і перетворень – джерело інформації, мета перетворення і місце призначення.

10. Журнали зберігаються в брокері повідомлень. Це дозволяє швидко визначати причини збоїв, відновлювати дані та виправляти помилки без повторного виникнення проблеми.

Порівняльний аналіз ESB

Нижче розглянемо, як компоненти *ESB* реалізовані в рішеннях, які найчастіше пропонуються на казахстанському ринку (*Talend* [9], *Mule* [10], *WSO*2 [11], *Red Hat Fuse*) (табл. 1).

Таблиця 1. Порівняльний аналіз *ESB*

№	Критерій / *ESB*	*Talend*	*Mule*	*WSO*2	*Red Hat Fuse*
1	Наявність студії	+	+	+	±
2	Підтримка брокера повідомлень	*JMS* 1.1, *Microsoft MQ* 3.0, *JBoss Messaging* 1.4.4, *IBM MQ* 8.0, *Apache ActiveMQ* 5.13.2	*Anypoint MQ, IBM MQ, Apache Kafka, JMS* 1.0.2, 1.1, 2.0 *support*	*Amazon SQS, JMS support, Apache Kafka*	*Apache ActiveMQ, Apache Kafka, AWS MQ, RabbitMQ, JMS support*
3	Логування	статистика виконання завдань і компонентів, помилок, попереджень і винятків на рівні завдань, потоків даних всередині завдань; логування в *Elastic, Apache Log4j, Apache Commons Logging, Trace Logs*	логування в межах кожної інтеграції, створеної в *Mule*: помилки та події, обов'язкові для логування за логікою інтеграції; логування запуску, зупинки, розгортання та відключення сервісів та інтеграцій *Mule*	Логування на основі *Apache Log4j* за допомогою бібліотеки *Apache Commons Logging*. Події системи та компонентів логеруються окремо	Логування на основі *Apache Log4j* через бібліотеку *Apache Commons Logging, SLF4J, java.util.logging, Elastic*
4	Моніторинг	+	+	+	+

Розробники також додають до цього списку *Apache Kafka, Kafka Connect* та *RabbitMQ*, але ці два рішення не є *ESB*, і розглядати їх в рамках цього аналізу недоцільно. В якості критеріїв виберемо основні функціональні компоненти шин даних: наявність студії, підтримка брокера повідомлень, спосіб логування та моніторингу.

Впровадження ESB для ефективного функціонування SIEM систем на об'єктах КІ

Розроблена платформа [3, 12] використовує механізм кореляції подій безпеки, що робить її спеціалізованою та повнофункціональною *SIEM* системою. Платформа забезпечує кореляцію

нормалізованих журналів/подій, пошук/запити для аналізу загроз та джерел інформації про вразливості, а також генерує тривоги з урахуванням ризиків. Основна її мета – збирати якнайбільше подій з інфраструктури організації [13].

Під час цифрової трансформації компанії, незалежно від їх розміру, використовують кілька інформаційних систем, які оперують масивами даних, що перетинаються. Обмін даними відбувається через *ESB* з використанням різних протоколів і форматів, що дозволяє уникнути модифікації систем, які інтегруються. Використання *ESB* для систем класу *SIEM* спрямоване на збалансований розподіл навантаження на сервіси та забезпечення безпеки обміну даними.

Давайте розглянемо, як сервіси можуть безпосередньо взаємодіяти один з одним. Отримати дані з додатку може потребувати складного багаторівневого ланцюжка операцій. Таких сервісів може бути від декількох до десятків або сотень. Постійний обмін повідомленнями між системами може створити велике навантаження, що призводить до тривалих затримок і постійних збоїв у роботі додатків (рис. 6). Також варто зазначити, що зміни або оновлення однієї системи можуть неминуче вплинути на всі інші сервіси, які від неї залежать.

Рис. 6. Взаємодія служб без шини

Використання *ESB* для систем класу *SIEM* повністю змінює організацію процесів в компанії. А саме, додаткам більше не потрібно спілкуватися безпосередньо один з одним, замість цього кожен з них взаємодіє тільки з інтеграційною платформою. Це миттєво усуває необхідність у величезній кількості методів доступу, стільки інтерфейсів, скільки буде потрібно сервісів. Якщо необхідно внести зміни в одну з систем, це не вплине на інші корпоративні додатки. *ESB* одноосібно візьме на себе всі ці завдання (рис. 7).

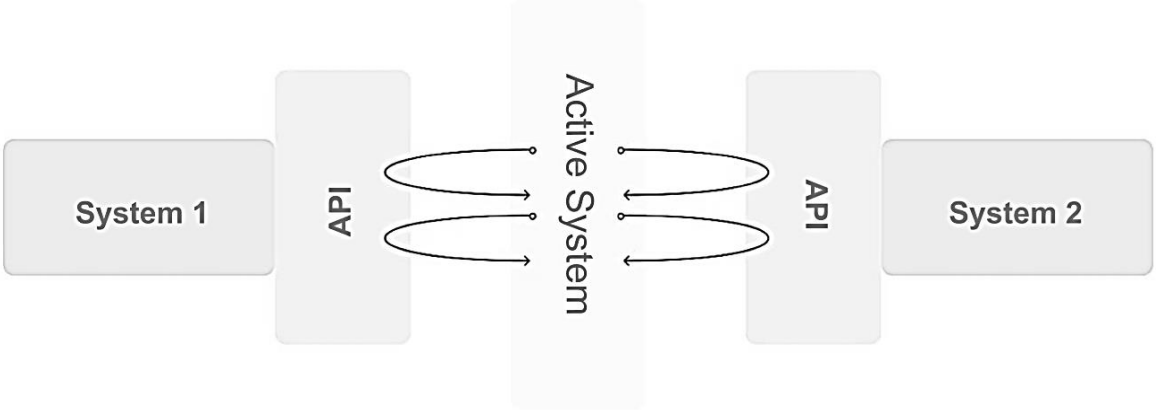

Рис. 7. Взаємодія служб з шиною

Таким чином, цей підхід, на відміну від традиційної архітектури *Point-to-point* (де сервіси взаємодіють безпосередньо один з одним), має більшу гнучкість. Сценарії інтеграції можуть бути змінені з мінімальним втручанням розробника. Переваги рішення: спрощення інтеграції додатків за рахунок впровадження *ESB* для систем класу *SIEM* економить час і ресурси, покращує функціонування сервісів, підвищує ефективність і безпеку організації.

Для збору інформації (подій) система використовує своїх агентів, які встановлюються в контрольованих підсистемах, а також стандартні існуючі механізми збору подій (*syslog, snmp* тощо). Для контролю мережі може використовуватися приймач *NetFlow*, що працює від мережевого обладнання. Він також може використовуватися для аналізу мережевого трафіку шляхом дзеркального відображення від мережевого обладнання, або шляхом пересилання трафіку через себе

Система надсилає події зашифрованим каналом до брокера повідомлень. Якщо немає зв'язку з ним, то система забезпечує тимчасове зберігання даних, мінімізуючи ризики втрати критично важливої інформації. У системі, що контролюється, може бути встановлено кілька брокерів. Основні принципи роботи:

- Під брокером повідомлень слід розуміти спеціальне програмне забезпечення, яке забезпечує гарантовану доставку повідомлень з декількох джерел декільком одержувачам. Це електронна черга для повідомлень.
- Репозиторій – це спеціальне сховище необроблених записів у зашифрованому вигляді. Важлива частина для збору юридично значущих доказів для розслідування інцидентів.

Горизонтально розширювані бази даних є відмінною архітектурною перевагою розробленої платформи. Система використовує розподілені бази даних різних типів для паралельного вирішення завдань контролю метрик (моніторингу) та контролю подій (*SIEM*) [14]:

- Висока швидкість обробки великих потоків інформації.
- Мінімальні затримки при обробці даних.
- Мінімальні затримки при побудові аналітичних звітів та запитів.
- Висока відмовостійкість.
- Можливість розширення сховища шляхом додавання вузлів без простою бази даних.

Модуль моніторингу – це комплексне програмне забезпечення для контролю метрик. Зазвичай такі завдання називаються "Моніторинг" і полягають у відстеженні кількісних показників систем у реальному часі. Коли метрики потрапляють у зони ризику, модуль створює подію порушення безпеки. Цей модуль має інтерактивний графічний інтерфейс.

Модуль аналітики – це комплексне програмне забезпечення для аналізу подій, яке виконує нормалізацію, кореляцію та аналіз подій. Він також знаходить залежності, визначає подію як інцидент та інформує інші системи. Крім того, він має інтерактивний графічний інтерфейс.

Сервіс-орієнтована архітектура, частиною якої є розроблена платформа, об'єднує всі *API*, що забезпечує наскрізну інтеграцію. *API* – це так званий набір правил та умов для спілкування програм між собою: вхідні та вихідні дані, типи операцій. Використання *API* значно спрощує взаємодію: він пов'язує можливості різних сервісів, формуючи інтерфейси, доступні для різних користувачів (рис. 8).

Мікросервісна архітектура відрізняється від традиційного *ESB*-підходу для систем *SIEM*-класу тим, що її функції організовані в невеликі сервіси, кожен з яких відповідає за окрему задачу, підтримується однією командою і може працювати ізольовано від інших. При такому підході немає централізованої бази. Кожен сервіс має своє сховище інформації. Однак *ESB* для систем класу *SIEM* виконує лише функцію транспорту, будучи, по суті, лише брокером повідомлень. Взаємодія між користувачем та сервісами платформи також здійснюється через *API* [15].

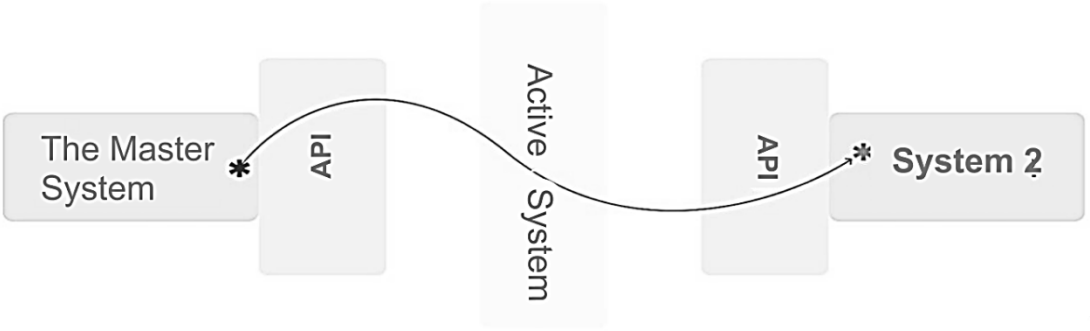

Рис. 8. Взаємодія *SOA*-сервісу з шиною

Специфікація SIEM

З урахуванням [2-4, 16-17] специфікація *SIEM* систем може бути відображена у вигляді основних та додаткових вимог:

1. Основні вимоги:

1.1. Системи спеціального призначення.

SIEM призначена для моніторингу та аналізу подій ІБ і повинна:

- здійснювати централізований збір, зберігання та обробку подій системних журналів (логів), а також мережевих потоків з різних систем інфраструктури Замовника;
- виділяти в загальному масиві даних важливі події та інциденти ІБ, що повинно дозволити фахівцям з ІБ Замовника, сконцентруватися на найбільш серйозних інцидентах і своєчасно реагувати на них;
- інформувати персонал Замовника про виявлені інциденти інформаційної безпеки шляхом надсилання повідомлень на електронну пошту.

1.2. Системи з централізованим управлінням.

SIEM повинна забезпечувати централізоване управління всіма своїми компонентами і функціоналом через єдиний графічний веб-інтерфейс.

1.3. Візуалізація даних (Dashboards). SIEM:

- дозволяє створювати графічні панелі (дашборди) за будь-якими подіями, з автоматичним оновленням із заданим інтервалом;
- підтримує створення нових графічних панелей або модифікацію існуючих за допомогою "майстра", методом, що не вимагає використання мов програмування;
- дозволяє зберігати графічні панелі для колективного використання. Графічні панелі повинні підтримувати різні типи представлення даних: таблиці, кругові та лінійні діаграми тощо, вони повинні функціонувати автоматично, без необхідності регулярного обслуговування оператором;
- підтримка відображення графічних панелей через *WEB*;
- підтримка інтерфейсу.

1.4. Підтримка API:

повинна мати відкритий програмний інтерфейс *API* для можливості інтеграції з іншими модулями.

1.5. Підтримка автентифікації та авторизації.

SIEM повинна підтримувати наступні методи для забезпечення автентифікації та авторизації користувачів:

- токени (для доступу до *API*);
- локальна база користувачів;
- *Active Directory*;
- *LDAP*.

1.6. Підтримка оновлень.

SIEM повинна підтримувати можливість автоматичного та/або ручного оновлення по мірі виходу нових версій.

1.7. Відмовостійкість.

База даних *SIEM* повинна підтримувати кластерну організацію в кількості не менше двох вузлів (*node*).

1.8. Масштабування.

SIEM:

- забезпечує горизонтальне масштабування шляхом додавання обладнання та, за необхідності, придбання додаткових

ліцензій на *SIEM* відповідно до чинної (на момент масштабування) політики ліцензування;

- має компонент зберігання подій (базу даних), в якому реалізовані наступні функції:
- Масштабування без фіксованого ліміту на обсяг зберігання подій (додавання додаткового обладнання при необхідності).
- Відмовостійка реалізація.

1.9. Збір та фільтрація подій.

SIEM:

- підтримує стандартні методи збору логів подій: *Syslog, Raw*/*Plaintext, GELF, CEF*, файлові журнали подій (за допомогою агентів для *Linux*/*Windows*);
- підтримує аналіз подій у реальному часі;
- забезпечує фільтрацію, а також відображення через користувацький інтерфейс події в реальному часі, де користувач може одразу застосувати фільтри;
- зберігає критерії пошуку для швидкого доступу;
- підтримує пошук за подіями з використанням мови запитів (якщо ви використовуєте власну мову запитів, вона має бути описана в документації);
- надає користувачеві можливість самостійно підключати джерела подій, які не підтримуються за замовчуванням;
- підтримує передачу даних від джерел до системи керування захищеним каналом (якщо в протоколі є підтримка захищеної передачі);
- підтримує централізоване управління агентами через інтерфейс *SIEM* (для агентів сімейства *Beats*).

1.10. Вимоги до управління акаунтом.

SIEM:

- підтримує рольову модель управління із заздалегідь визначеним набором ролей;
- має можливість створювати та використовувати групи користувачів (*Teams*);
- має систему управління токенами для авторизації в *API*.

1.11. Вимоги до кількості подій в секунду (EPS):

- Середньоденний - не більше 200 *EPS*;
- Максимальний у найбільш завантажену годину - не більше 400 *EPS*.

1.12. Вимоги до інформації, що зберігається в базі.

Щоденний обсяг інформації, що зберігається в базі даних подій, становить не більше 4 ГБ на добу; Термін зберігання подій у базі даних та/або архіві *SIEM* - не менше 3 років.

2. Додаткові вимоги:

2.1. Постачальник забезпечує встановлення програмного забезпечення *SIEM* на фізичних серверах та/або платформі віртуалізації Замовника.

2.2. Постачальник надає Замовнику розрахунок потреб у ресурсах для встановлення *SIEM*. Розрахунок здійснюється Постачальником на основі вимог до продуктивності, зазначених у цій специфікації

2.3. Замовник забезпечує виділення ресурсів відповідно до зазначеного розрахунку, встановлення та базове налаштування операційних систем (включаючи налаштування дискової підсистеми та мережевих інтерфейсів) для встановлення *SIEM*. Постачальник надає Замовнику дистрибутив операційної системи (на носії або у вигляді посилання для завантаження) та, за необхідності, ліцензію на операційні системи.

2.4. Замовник забезпечує доступ з серверів до мережі Інтернет на час встановлення та налаштування *SIEM*.

2.5. На весь період експлуатації *SIEM* Замовник забезпечує постійний доступ з серверів *SIEM* до сервера ліцензування для контролю ліцензії.

2.6. Постачальник протягом десяти календарних днів завершує встановлення та налаштування системи *SIEM*.

Отже, використання запропонованої *ESB* шини для ефективного функціонування *SIEM* систем надасть численні переваги, такі як широкий спектр роз'ємів і

масштабованість рішення, гнучка маршрутизація даних, гарантована доставка інформаційних повідомлень, організація захищеного каналу передачі, централізоване управління, можливість моніторингу та діагностики стану передачі, а також можливість інтеграції зі сторонніми чергами повідомлень.

Висновки

У роботі проведено аналіз сучасних рішень *ESB* та з'ясовано, що кожен продукт має свої особливості, які визначають їх сферу використання. А саме, для компаній, які шукають безкоштовні застосунки, використання рішення *Fuse* є найкращим варіантом, але на початку роботи потребує додаткових налаштувань. *Talend* або *Mule* будуть хорошим варіантом на ранніх стадіях розвитку компанії. *WSO2* є оптимальним варіантом з високою функціональністю та простотою розрахунку вартості ліцензії.

Реалізовано та впроваджено *ESB* для ефективного функціонування *SIEM* систем на об’єктах КІ. Розроблено *ESB* на базі сервіс-орієнтованої архітектури. Платформа використовує розподілені бази даних різних типів для розв'язання паралельних завдань із контролю метрик і подій. Це на порядок збільшує параметри, забезпечуючи: швидкість опрацювання великих потоків інформації; мінімальні затримки на опрацювання даних; мінімальні затримки для побудови аналітичних звітів і запитів; високу відмовостійкість; розширюваність сховища шляхом простого додавання вузлів без простою бази. Використання *API* значно спрощує взаємодію, об’єднуючи можливості різних сервісів, та утворюючи доступні різним користувачам інтерфейси.

Також, було сформовано відповідну специфікацію реалізації *SIEM* систем на об’єктах КІ.

Література

1. Skendžić A., Kovačić B., Balon B. Management and Monitoring Security Events in a Business Organization – SIEM system. *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)* / Opatija, Croatia, 2022. P. 1203–1208.

2. Gnatyuk S., Berdibayev R., Fesenko A., Kyryliuk O., Bessalov A. Modern SIEM Analysis and Critical Requirements Definition in the Context of Information Warfare. *CEUR Workshop Proceedings*. 2021. Vol. 3188. P. 149–166.

3. Berdibayev R., Gnatyuk S., Yevchenko Yu., Kishchenko V. A concept of the architecture and creation for SIEM system in critical infrastructure. *Studies in Systems, Decision and Control*. 2021. Vol. 346. P. 221–242.

4. Gnatyuk S., Berdibayev R., Azarov I., Baisholan N., Lozova I. Modern Types of Databases for SIEM System Development. *CEUR Workshop Proceedings*. 2021. Vol. 3187. P. 127–138.

5. Jin Z., Zhu H. A Framework for Agent-Based Service-Oriented Modelling. *2008 IEEE International Symposium on Service-Oriented System Engineering* / Jhongli, Taiwan, 2008. P. 160–165.

6. Li W. Design and Implementation of Software Testing Platform for SOA-Based System. *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)* / Chengdu, China, 2021. P. 1094–1098.

7. ESB (Enterprise Service Bus). URL: https://www.ibm.com/cloud/learn/esb.

8. Dai P. Design and implementation of ESB based on SOA in power system. *2011 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)* / Weihai, China, 2011. P. 519–522.

9. Sreemathy J., Joseph I.V., Nisha S., Prabha C.I., Priya G.R.M. Data Integration in ETL Using TALEND. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* / Coimbatore, India, 2020. P. 1444–1448.

10. Mkhwanazi X., Le H., Blake E. Clustering between Data Mules for Better Message Delivery. *2012 26th International Conference on Advanced Information Networking and Applications Workshops* / Fukuoka, Japan, 2012. P. 209–214.

11. Kumara I., Gamage C. Towards Reusing ESB Services in Different ESB Architectures. *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops* / Seoul, Korea (South), 2010. P. 25–30.

12. Gnatyuk S., Berdibayev R., Smirnova T., Avkurova Z., Iavich M. Cloud-Based Cyber Incidents Response System and Software Tools. *Communications in Computer and Information Science*. 2021. Vol. 1486. P. 169–184.

13. Laue T., Kleiner C., Detken K.-O., Klecker T. A SIEM Architecture for Multidimensional Anomaly Detection. *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* / Cracow, Poland, 2021. P. 136–142.

14. A*sef P*., *Taheri R*., *Shojafar M*., *Mpora*s *I*., *Tafazolli R*. *SIEMS*: A *Secure Intelligent Energy Management System for Industrial IoT applications*. *IEEE Transactions on Industrial Informatics*. 2022. *P*. 1–12.

15. Orsós M., Kecskés M., Kail E., Bánáti A. Log collection and SIEM for 5G SOC. *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)* / Poprad, Slovakia, 2022. P. 000147–000152.

16. Gnatyuk S., Berdibayev R., Sydorenko V., Polozhentsev A., Ryabyy M. Enterprise Service Bus Construction in SOA Architecture for SIEM Implementation in Critical Information Infrastructure. *CEUR Workshop Proceedings*. 2022. Vol. 3288. Paper 2. P. 11–20.

17. Gnatyuk S., Berdibayev R., Sydorenko V., Berdibayeva G., Yudin O. Methodological Bases of Critical Information Infrastructure Identification and Security Assessment: Monograph. Kyiv : "Pro Format" Publishing House, 2023. 129 p.

Гнатюк С.О., Бердибаєв Р.Ш., Богун А.М., Сидоренко В.М., Положенцев А.А., Жигаревич О.К.

ІНТЕГРАЦІЙНА ШИНА ДАНИХ ДЛЯ ЕФЕКТИВНОГО ФУНКЦІОНУВАННЯ СИСТЕМИ УПРАВЛІННЯ ПОДІЯМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Кількість кіберзагроз у сфері ІКТ постійно збільшується, тому розробка нових засобів для забезпечення безпеки є дуже важливим і актуальним науковим завданням. Серед таких засобів варто відзначити SIEM системи, які спрямовані на аналіз подій та управління інцидентами з метою запобігання негативним наслідкам та зменшення шкоди від кіберзагроз для користувачів. У попередніх дослідженнях автори провели аналіз існуючих SIEM систем та типів баз даних для них, а також розробили нову архітектуру хмарної SIEM системи. Наступним кроком є дослідження Enterprise Service Bus (ESB). У статті було визначено роль ESB у концепції архітектури Service-Oriented Architecture (SOA), визначені її функції та переваги. Також автори проаналізували найпопулярніші сучасні рішення ESB та надали рекомендації щодо впровадження розвиненої SIEM системи на об'єктах критичної інфраструктури. Розроблений ESB компонент для ефективного функціонування SIEM систем на об'єктах КІ забезпечить цілу низку переваг, такі як широкий спектр роз'ємів і масштабованість рішення, гнучка маршрутизація даних, гарантована доставка інформаційних повідомлень, організація захищеного каналу передачі, централізоване управління, можливість моніторингу та діагностики стану передачі, а також можливість інтеграції зі сторонніми чергами повідомлень. Крім того, у дослідженні було сформовано специфікацію для SIEM системи в критичній інфраструктурі.

Ключові слова: *SIEM, управління інцидентами, ESB, кіберзагрози, хмарна архітектура, SOA.*

Gnatyuk S.O., Berdibayev R.Sh., Bogun A.M., Sydorenko V.M., Polozhentsev A.A., Zhyharevych O.K.

ENTERPRISE SERVICE BUS FOR EFFICIENT FUNCTIONING OF THE INFORMATION SECURITY EVENT MANAGEMENT SYSTEM

The number of cyber threats in ICT is increasing and the development of new security oriented instrumental tools is very important and relevant scientific task. SIEM systems are category of such tools, directed on log analysis and incident management to prevent negative consequences minimize damage of cyber threats for end user. In the previous works authors have analyzed existed SIEM systems and DB types for them as well as created new architecture of cloud-based SIEM. Next step of this research project is ESB justification. The paper defines the place of ESB distributed data bus in the concept of SOA architecture, identifies the functions and benefits. Also authors analyzed most popular up-to-date ESB solutions and provides recommendations in context of developed SIEM implementation in the critical infrastructure. The developed ESB component for the effective functioning of SIEM systems at CI facilities will provide a number of advantages, such as a wide range of connectors and solution scalability, flexible data routing, guaranteed delivery of information messages, organization of a secure transmission channel, centralized management, the ability to monitor and diagnose transmission status, as well as the possibility of integration with third-party message queues. Besides, the data sheet for SIEM in critical infrastructure was formed and proposed in this paper.

Keywords: *SIEM, incident management, ESB, cyber threat, cloud-based architecture, SOA.*